



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/090,041	03/01/2002	Reinhard P. Klemm	Klemm 037	6798

48289 7590 12/08/2005

COHEN, PONTANI, LIEBERMAN & PAVANE
551 FIFTH AVENUE
SUITE 1210
NEW YORK, NY 10176

EXAMINER

MANOSKEY, JOSEPH D

ART UNIT PAPER NUMBER

2113

DATE MAILED: 12/08/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No. 10/090,041	Applicant(s) KLEMM, REINHARD P.	
	Examiner Joseph D. Manoskey	Art Unit 2113	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 21 September 2005.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-5, 7-23 and 26-43 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-5, 7-23 and 26-43 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 01 March 2002 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
| 3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date <u>9/21/05</u> . | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

Claim Rejections - 35 USC § 103

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

2. Claims 1-5, 7-23, and 26-43 are rejected under 35 U.S.C. 103(a) as being unpatentable over Klemm and Singh, *Enhancing Java Server Availability with JAS*, published online 16 March 2001, hereinafter referred to as "Reinhard", in view of Shi et al., U.S. patent 6,757,897, hereinafter referred to as "Shi".

3. Referring to claim 1, Reinhard teaches a method executing a program of the target application (See page 698). Also, Reinhard teaches receiving exceptions, this is interpreted as receiving a notification of an event specifying an unexpected or erroneous behavior in execution of said program code (See page 701). Reinhard teaches events and actions are expressed in a configuration file for determining the action to be performed in response to the event, including restarting an idle application, this is interpreted as searching a configuration file that maintains a listing of events and associated actions, at least one of said actions being a restarting said target application

when said target application becomes idle; retrieving an action from said configuration file that is associated with said event; and carrying out said action (See page 702).

Reinhard does not teach said configuration file specifies periodic checking for a thread starvation condition, however does Reinhard express a desire to include detecting thread starvation (See page 716). Shi teaches scheduling a task to prevent task or process starvation conditions to run often (See Col. 16, line 67 to Col. 17, line 5). Shi also teaches setting an event to occur after predetermined elapsed time and if the task is still running after the elapsed time, generating a yield signal to allow another task to run (See Col. 4, line 60 to Col. 5 line 5). It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the scheduling of the task to prevent starvation conditions of Shi with the configuration file of Reinhard. This would have been obvious to one of ordinary skill in the art at the time of the invention to do because it allows other tasks perform in place of one that is causing the starvation condition (See Shi, Col. 16, line 67 to Col. 17, line 5).

4. Referring to claim 2, Reinhard and Shi teach all the limitations (See rejection of claim 1) including the configuration file having a plurality of events and actions including thread restarts (See Reinhard, page 702).

5. Referring to claim 3, Reinhard and Shi teach all the limitations (See rejection of claim 1) including the configuration file having a plurality of events and discloses the method is including the ability to visualize some aspects of the state of the target

application, this interpreted as configuration file includes a plurality of events and associated actions, where said actions are taken from a set that includes a partial state dump (See Reinhard, page 699 and 702).

6. Referring to claim 4, Reinhard and Shi disclose all the limitations (See rejection of claim 1) including the configuration file having a plurality of events, actions including application restarts and including checkpointing, this is interpreted as said configuration file includes a plurality of events and associated actions, where said actions are taken from a set that restarting said target application from a checkpointed state (See Reinhard, page 700 and 702).

7. Referring to claim 5, Reinhard and Shi teach all the limitations (See rejection of claim 1) including the configuration file having a plurality of events, actions including thread restarts and including checkpointing, this is interpreted as said configuration file includes a plurality of events and associated actions, where said actions are taken from a set that restarting a thread of said target application from a checkpointed state (See Reinhard, page 700 and 702).

8. Referring to claim 7, Reinhard and Shi teach all the limitations (See rejection of claim 1) including the method including detecting thread starvation, this is interpreted as a step of periodically checking for the thread starvation condition (See Reinhard, page 716 and Shi Col. 16, line 67 to Col. 17, line 5).

9. Referring to claim 8, Reinhard and Shi teach all the limitations (See rejection of claim 7) including the method including detecting thread starvation and determining if a thread is hung, i.e. execution time exceeds user specification, this is interpreted as said checking for thread starvation condition includes the step of checking whether there is a subset of threads of said target application that take more than predetermined share of CPU time of said computer (See Reinhard, page 701 and 716 and Shi Col. 16, line 67 to Col. 17, line 5).

10. Referring to claim 9, Reinhard and Shi teach all the limitations (See rejection of claim 7) including the method including detecting thread starvation and determining if a thread is hung, i.e. execution time exceeds user specification, thus other threads are receiving less time, this is interpreted as said checking for thread starvation condition includes the step of checking whether a thread of said target application receives less than a predetermined share of CPU time of said computer (See Reinhard, page 701 and 716 and Shi Col. 16, line 67 to Col. 17, line 5).

11. Referring to claim 10, Reinhard and Shi teach all the limitations (See rejection of claim 7) including the method including detecting thread starvation and determining if a thread is hung, i.e. execution time exceeds user specification. The length of time a hung thread runs is the length of time a starved thread is not executing. This is interpreted as recording a time t_1 for relinquishing CPU time of said computer;

relinquishes CPU time of said computer for a specified time, records a time t2 when it reacquires CPU time of said computer, and concludes that the thread starvation condition exists when time t2 is greater than time t1 by a predetermined amount (See Reinhard, 701 and 716 and Shi Col. 16, line 67 to Col. 17, line 5).

12. Referring to claim 11, Reinhard and Shi teach all the limitations (See rejection of claim 8) including the configuration file having a plurality of events, actions including thread restarts and including thread starvation, this is interpreted as said configuration file includes a plurality of events and associated actions, where said actions are taken from a set that includes a recovery from the thread starvation condition (See Reinhard, page 702 and 716 and Shi Col. 16, line 67 to Col. 17, line 5).

13. Referring to claim 12, Reinhard and Shi teach all the limitations (See rejection of claim 11) including detecting thread starvation and suspend threads to deal with resource or performance penalties, this is interpreted as recovery from the thread starvation condition comprises suspending one or more threads for a preselected period of time (See Reinhard, page 705 and 716 and Shi Col. 16, line 67 to Col. 17, line 5).

14. Referring to claim 13, Reinhard and Shi teach all the limitations (See rejection of claim 12) including detecting thread starvation and suspend threads to deal with resource or performance penalties, this is interpreted as suspending one or more threads is carrying out by iteratively selecting a thread, suspending the selected thread,

and testing effect of the suspension of the selected thread on said thread starvation condition (See Reinhard, page 705 and 716 and Shi Col. 16, line 67 to Col. 17, line 5).

15. Referring to claim 14, Reinhard and Shi teach all the limitations (See rejection of claim 12) including detecting thread starvation and suspend threads to deal with resource or performance penalties, this is interpreted as said one or more threads that are suspended are threads that eliminate said thread starvation condition, or reduce the thread starvation condition by a predetermined amount (See Reinhard, page 705 and 716 and Shi Col. 16, line 67 to Col. 17, line 5).

16. Referring to claim 15, Reinhard and Shi teach all the limitations (See rejection of claim 1) including the configuration file having a plurality of events, actions including application restarts and including checkpointing, this is interpreted as a step checkpointing said target application in accordance with a predetermined algorithm, and said configuration file includes at least one action that restarts said target application based on information obtained via said checkpointing (See Reinhard, page 700 and 702).

17. Referring to claim 16, Reinhard and Shi teach all the limitations (See rejection of claim 1) including the configuration file having a plurality of events, actions including checkpointing and providing actions to be made after the application events have reached a threshold value, this is interpreted as a step checkpointing said target

application that is executed at regular intervals, or when the amount of information stored for said target application exceeds a predetermined threshold, or when activity level of said target application exceeds a predetermined threshold (See Reinhard, page 700 and 702).

18. Referring to claim 17, Reinhard and Shi teach all the limitations (See rejection of claim 1) including the configuration file having a plurality of events, actions including restarts and checkpointing, this is interpreted as said configuration file includes a plurality of events and associated actions, where said actions are taken from a set that includes a checkpointing and a restart based on information obtained from said checkpointing (See Reinhard, page 700 and 702).

19. Referring to claim 18, Reinhard and Shi teach all the limitations (See rejection of claim 17) including the method including checkpointing, this is interpreted as said checkpointing is in accordance with information provided by said program code (See Reinhard page 700 and 702).

20. Referring to claim 19, Reinhard and Shi teach all the limitations (See rejection of claim 18) including the method being able to specify protocols for certain classes of threads, this is interpreted said program code specifies thread and all progeny of said thread (See Reinhard, page 703).

Art Unit: 2113

21. Referring to claim 20, Reinhard and Shi teach all the limitations (See rejection of claim 1) including the configuration file having a plurality of events, actions including restarts and checkpointing, this is interpreted as a step of checkpointing when said notification of said event is received, and said configuration file includes at least one action that restarts said target application based on information obtained via said checkpointing (See Reinhard, page 700 and 702).

22. Referring to claim 21, Reinhard and Shi teach all the limitations (See rejection of claim 20) including the configuration file having a plurality of events, actions including checkpointing and discloses the method including the ability to visualize some aspects of the state of the target application, this is interpreted as said checkpointing stores state information in accordance with specification by said target application (See Reinhard, page 699, 700 and 702).

23. Referring to claim 22, Reinhard teaches executing a program of the target application with an application supervisor, this is interpreted as a system including a processing unit including a target application executing on said processing unit and a supervisor software module executing on said processing unit (See page 698).

Reinhard discloses the supervisor being external to the application, this is interpreted as execution code of said target application is unaware of said supervisor module (See page 701). Also, Reinhard teaches receiving exceptions and Reinhard teaches events and actions are expressed in a configuration file for determining the action to be

performed in response to the event, including restarting an idle application and quitting the application, this is interpreted as the supervisor module monitors execution of said target application and in response to an error or unexpected behavior in said execution takes action that affects said execution which action is taken from a set of actions that includes an action for terminating execution of said software module and at least an action that restarts said target application only when the target application becomes idle(See pages 701 and 702). Reinhard teaches determining if the error exists, such as if a thread is hung, by determining if the execution time has exceeded a user specification, (See page 701).

Reinhard does not teach determining whether said error exists by checking implicit conditions including at least a thread starvation condition in said execution of said software module, however does Reinhard express a desire to include detecting thread starvation (See page 716). Shi teaches scheduling a task to prevent task or process starvation conditions to run often (See Col. 16, line 67 to Col. 17, line 5). Shi also teaches setting an event to occur after predetermined elapsed time and if the task is still running after the elapsed time, generating a yield signal to allow another task to run (See Col. 4, line 60 to Col. 5 line 5). It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the scheduling of the task to prevent starvation conditions of Shi with the supervisor module of Reinhard. This would have been obvious to one of ordinary skill in the art at the time of the invention to do because it allows other tasks perform in place of one that is causing the starvation condition (See Shi, Col. 16, line 67 to Col. 17, line 5).

24. Referring to claim 23, Reinhard and Shi teach all the limitations (See rejection of claim 22) including the configuration file having a plurality of events and actions including thread restarts (See Reinhard, page 702).

25. Referring to claim 26, Reinhard and Shi teach all the limitations (See rejection of claim 22) including the configuration file having a plurality of events and discloses the system including the ability to visualize some aspects of the state of the target application, this is interpreted as where said set includes storing a file a partial state of said target application (Sees Reinhard, page 699 and 702).

26. Referring to claim 27, Reinhard and Shi disclose all the limitations (See rejection of claim 22) including the system including the ability to visualize some aspects of the state of the target application and the supervisor dealing with threads of the application, this is interpreted as where said set includes storing in a file state information of thread of said target application (See Reinhard, pages 699 and 702).

27. Referring to claim 28, Reinhard and Shi disclose all the limitations (See rejection of claim 22) including the configuration file having a plurality of events and actions including checkpointing, this is interpreted as where said set includes checkpointing (See Reinhard, page 700 and 702).

28. Referring to claim 29, Reinhard and Shi disclose all the limitations (See rejection of claim 28) including the configuration file having a plurality of events and actions including checkpointing, this is interpreted as where said checkpointing is triggered by code included in said target application (See Reinhard, page 700 and 702).

29. Referring to claim 30, Reinhard and Shi teach all the limitations (See rejection of claim 23) including taking actions that are triggered by specified number of a events exceeding a maximum, this is interpreted as where said set further includes taking no action relative to execution to said software module, but incrementing an error counter (See Reinhard, page 703). Reinhard teaches taking action when the number of events exceed a maximum, ignore event, quit application (this is interpreted to also include terminating threads, restart application immediately, restart idle application, restart thread (See Reinhard, page 702).

30. Referring to claim 31, Reinhard and Shi teach all the limitations (See rejection of claim 22) including receiving exceptions (See Reinhard, page 701). Reinhard also teaches events and actions are expressed in a configuration file for determining the action to be performed in response to the event, this is interpreted as where said action taken by said supervisor is retrieved from a configuration file that is specific to said target applications, which file specifies actions to be taken in response to specified signal reporting on said error or unexpected behavior (See Reinhard, page 702).

31. Referring to claim 32, Reinhard and Shi disclose all the limitations (See rejection of claim 31) including the use tailoring configuration to varying degrees, this is interpreted as said configuration file is modifiable by a user of said application module (See Reinhard, page 702).

32. Referring to claim 33, Reinhard and Shi disclose all the limitations (See rejection of claim 31) including the supervisor generating a default configuration from the target bytecode, this is interpreted as further comprising a configuration manager that creates said configuration file by evaluating said code of said application module (See Reinhard, page 702).

33. Referring to claim 34, Reinhard and Shi teach all the limitations (See rejection of claim 22) including the system having checkpointing and restarting the application, this is interpreted as said supervisor interacts with a checkpointing module before taking an action from said set that restarts execution of said application module (See Reinhard, page 700 and 702).

34. Referring to claim 35, Reinhard and Shi disclose all the limitations (See rejection of claim 34) including the system having checkpointing and being an external supervisor to the application, this is interpreted as said checkpointing module that checkpoints execution is independent of said supervisor (See Reinhard, pages 700 and 701).

35. Referring to claim 36, Reinhard and Shi teach all the limitations (See rejection of claim 35) including the configuration file having a plurality of events, actions including application restarts and including checkpointing, this is interpreted as said independent module checkpoints pursuant to a predetermined algorithm of said checkpointing module (See Reinhard, page 700 and 702).

36. Referring to claim 37, Reinhard and Shi teach all the limitations (See rejection of claim 34) including the configuration file having a plurality of events, actions including checkpointing and restarting applications, this is interpreted as a configuration file that specifies errors that restart execution of said application with information developed by said checkpointing (See Reinhard, page 700 and 702).

37. Referring to claim 38, Reinhard and Shi teach all the limitations (See rejection of claim 34) including the configuration file having a plurality of events, actions including checkpointing and restarting applications or restarting applications immediately, this is interpreted as a configuration file that specifies errors that restart execution of said application with information developed by said checkpointing and errors that restart execution of said target application without information developed by said checkpointing (See Reinhard, page 700 and 702).

38. Referring to claim 39, Reinhard and Shi teach all the limitations (See rejection of claim 22) including the configuration file having a plurality of events, and actions that include restarting execution. Also Reinhard discloses the system including the ability to visualize some aspects of the state of the target application, this is interpreted as said supervisor maintains state information of an execution thread of said target application, which when said execution thread causes said checkpointing and errors that restart execution of said target application without information developed by said checkpointing (See Reinhard, page 699 and 702).

39. Referring to claim 40, Reinhard teaches executing a program of the target application with an application supervisor, this is interpreted as a system including a processing unit including an application software module executing on said processing unit and a supervisor software module executing on said processing unit (See page 698). Reinhard also teaches the supervisor being external to the application, the application sending exceptions to the supervisor, the system including the ability to visualize some aspects of the state of the target application, and checkpointing; this is interpreted as execution code of said software application module makes no reference to said supervisor module except for sending one or more messages to said supervisor, at one or more locations of said code, specifying a subset of state information of said application module for said supervisor module to keep track of possible checkpointing (See pages 699, 700, and 701). Reinhard teaches the system detecting events in the execution of the application and the system including the ability to visualize some

aspects of the state of the target application, this is interpreted as said supervisor module monitors execution of said application module, and concurrently keeps track of scope of said subset of state information specified in a most recently received one of said messages (See pages 699 and 700). Reinhard teaches determining if the error exists, such as if a thread is hung, by determining if the execution time has exceeded a user specification, (See page 701).

Reinhard does not teach determining whether said error exists by checking implicit conditions including at least a thread starvation condition in said execution of said software module, however does Reinhard express a desire to include detecting thread starvation (See page 716). Shi teaches scheduling a task to prevent task or process starvation conditions to run often (See Col. 16, line 67 to Col. 17, line 5). Shi also teaches setting an event to occur after predetermined elapsed time and if the task is still running after the elapsed time, generating a yield signal to allow another task to run (See Col. 4, line 60 to Col. 5 line 5). It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the scheduling of the task to prevent starvation conditions of Shi with the supervisor module of Reinhard. This would have been obvious to one of ordinary skill in the art at the time of the invention to do because it allows other tasks perform in place of one that is causing the starvation condition (See Shi, Col. 16, line 67 to Col. 17, line 5).

40. Referring to claim 41, Reinhard and Shi teach all the limitations (See rejection of claim 40) including the system having checkpointing and restarting the application, this

is interpreted as where said supervisor stores checkpointing information, pursuant to said scope of said subset of state information when said supervisor determines that execution of said application module is characterized by abnormal behavior that calls for restarting execution of said application module (See Reinhard, page 700 and 702).

41. Referring to claims 42, Reinhard and Shi teach all the limitations (See rejection of claim 41) including the system having checkpointing and restarting the application, this is interpreted as where, following said storing of checkpointing information, said supervisor restarts execution of said application module (See Reinhard, page 700 and 702).

42. Referring to claims 43, Reinhard and Shi teach all the limitations (See rejection of claim 40) including the supervisor being external to the application, the application sending exceptions to the supervisor and the system including the ability to visualize some aspects of the state of the target application, this is interpreted as where each of said one or more messages specifies said subset of state information by specifying one or more object trees (See Reinhard, pages 699, 700, and 701).

Response to Arguments

43. Applicant's arguments filed 21 September 2005 have been fully considered but they are not persuasive. The Applicant argues that Shi is directed to a method for attempting to avoid a thread starvation condition by using a task scheduling algorithm

and that there is no teaching in Shi of searching a configuration file which "specifies periodic checking for a thread starvation condition." The Examiner respectfully disagrees. Shi also teaches setting an event to occur after predetermined elapsed time and if the task is still running after the elapsed time, generating a yield signal to allow another task to run (See Col. 4, line 60 to Col. 5 line 5) and thus actively stops a task from creating a starvation condition and not just indirectly prevents them. The above rejections have been modified to include this clarification.

Conclusion

44. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).


A shortened statutory period for reply to this final action is set to expire **THREE MONTHS** from the mailing date of this action. In the event a first reply is filed within **TWO MONTHS** of the mailing date of this final action and the advisory action is not mailed until after the end of the **THREE-MONTH** shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than **SIX MONTHS** from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Joseph D. Manoskey whose telephone number is (571) 272-3648. The examiner can normally be reached on Mon.-Fri. (7:30am to 4pm).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Robert Beausoliel can be reached on (571) 272-3645. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

JDM
November 30, 2005


ROBERT BEAUSOLIEL
CATEGORY PATENT EXAMINER
TECHNOLOGY CENTER 2100